

基于改进 A* 算法的救援机器人路径规划研究

汤亚玲^{a,b}, 穆如如^a, 张学锋^a, 钟春燕^a

安徽工业大学 a. 计算机科学与技术学院; b. 特种重载机器人安徽省重点实验室, 安徽 马鞍山 243032

摘要:目的 化工园区救援机器人的路径规划是目前研究的热点, 针对救援机器人在路径规划中使用传统 A* 算法存在遍历冗余节点、内存消耗较大和运算速度较慢的问题, 提出一种改进的 A* 算法。方法 设计了自适应权重评价函数, 用于动态调整实际代价, 以提升算法的效率与收敛速度; 采用跳点搜索策略(Jump Point Search, JPS)来筛选跳点, 降低内存消耗和节点估计; 运用三次均匀 B 样条曲线对优化后的路径进行处理, 提高救援机器人前进的稳定性和可行性。结果 通过构建化工园区 Unity3D 仿真场景, 对救援机器人路径规划问题进行仿真研究, 仿真结果表明: 改进后的 A* 算法展现出较高的效率及准确性。结论 该算法为化工园区救援机器人的路径规划提供了一种更高效、更智能的解决方案, 满足化工园区路径快速规划的需求。

关键词:救援机器人; 跳点搜索策略; 路径规划; 自适应权重; 三次均匀 B 样条曲线

中图分类号:TP391.9 **文献标识码:**A **doi:**10.16055/j.issn.1672-058X.2025.0006.005

Research on Path Planning of Rescue Robots Based on Improved A* Algorithm

TANG Yaling^{a, b}, MU Ruru^a, ZHANG Xuefeng^a, ZHONG Chunyan^a

a. School of Computer Science and Technology; b. Anhui Province Key Laboratory of Special Heavy Load Robot, Anhui University of Technology, Ma'anshan 243032, Anhui, China

Abstract: Objective Path planning for rescue robots in chemical parks is a hot topic in current research. To address the issues of traversing redundant nodes, high memory consumption, and slow computation speed of the traditional A* algorithm in rescue robot path planning, an improved A* algorithm is proposed. **Methods** An adaptive weight evaluation function is designed to dynamically adjust the actual cost, so as to enhance the algorithm's efficiency and convergence speed. The jump point search strategy(JPS) is employed to screen jump points, reducing memory consumption and node estimation. A cubic uniform B-spline curve is used to process the optimized path to improve the stability and feasibility of the rescue robot's movement. **Results** The path planning problem of the rescue robot is simulated by constructing a Unity3D simulation scene in the chemical park, and the simulation results show that the improved A* algorithm exhibits high efficiency and accuracy. **Conclusion** This algorithm offers a more efficient and intelligent solution for the path planning of rescue robots in chemical parks, meeting the requirement for rapid path planning in such areas.

Keywords: rescue robot; jump point search strategy; path planning; adaptive weight; cubic uniform B-spline curve

1 引言

近年来机器人路径规划技术已经被广泛地应用于各个领域, 涉及日常生活、工业生产以及救援任务的众

多方面, 发挥着至关重要的作用。作为机器人导航的关键环节, 路径规划对于机器人的智能决策和优化行动具有关键意义。这项技术的不断发展和应用推动了

收稿日期: 2023-07-21 修回日期: 2023-09-19 文章编号: 1672-058X(2025)06-0033-08

基金项目: 特种重载机器人安徽省重点实验室开放课题(TZJQR007-2023)。

作者简介: 汤亚玲(1974—), 男, 安徽庐江人, 副教授, 硕士, 从事虚拟现实、数据挖掘、遗传算法和网络数据库系统研究。

通信作者: 张学锋(1978—), 男, 河北石家庄人, 教授, 博士, 从事虚拟现实技术与人工智能研究。Email: zxf_06@ahut.edu.cn.

引用格式: 汤亚玲, 穆如如, 张学锋, 等. 基于改进 A* 算法的救援机器人路径规划研究[J]. 重庆工商大学学报(自然科学版), 2025, 42(6): 33-40.

TANG Yaling, MU Ruru, ZHANG Xuefeng, et al. Research on path planning of rescue robots based on improved A* algorithm [J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2025, 42(6): 33-40.

机器人在各个领域的性能提升和任务执行效率的提高,同时也为人们的生活和工作带来了便利和改善。

工业救援机器人领域对路径规划技术有着迫切需求和重要性。在复杂的工业环境中,可能存在着大量的障碍物、狭窄的通道以及危险的工作条件,这为救援任务带来了极大的挑战。而工业救援机器人的任务是在紧急情况下迅速、准确地到达事故现场,进行物资搬运、危险区域搜索或者人员救援。为了确保工业救援机器人的高效运转和任务完成,路径规划必须具备快速响应、精准规划的特性。

在路径规划领域中,Dijkstra 算法、A* 算法、粒子群算法以及遗传算法等^[2]多种算法被广泛应用。这些算法均有各自不同的特点与优势。A* 算法以其卓越的搜索路径的稳定性及出色的实时性而著称,相较于粒子群算法和遗传算法等其他全局路径规划算法,它的实现效率更高,成为机器人应用中最广泛使用的全局路径规划算法之一。由于该算法具有良好的可扩展性与适应性,因此被广泛应用于各种环境中。然而,在复杂的场景中,A* 算法会遭遇大量冗余节点的搜索,从而导致算法效率的下降,因此对 A* 算法进行进一步的优化变得尤为紧迫。

针对上述问题,槐创锋等^[3]改进 A* 算法的搜索邻域为 7×7 ,一定程度上优化了搜索路径,但该方法增加了搜索的时间,降低搜索效率。吴鹏等^[4]提出了一种双向搜索策略,该策略能够同时从起点和目标点进行正反向路径搜索,从而显著提高搜索效率,但同时也增加了内部搜索过程中的资源消耗;孔继利等^[5]对算法的代价评估函数进行了改进,通过选择适当的权重系数,来降低算法扩展节点的数量,但是仍存在大量冗余节点,导致内存消耗和计算时间过长等问题。上述学者在传统 A* 算法的基础上进行改进,但改进策略均没有很好地解决 A* 算法在搜索过程中大量冗余节点的问题,本文设计了自适应权重评价函数,通过动态调整算法的搜索过程,并采用跳点搜索策略(JPS),进一步减少了内存开销和节点评估,从而减少了算法搜索过程中计算的节点数量。此外,在路径平滑方面采用三次均匀 B 样条曲线,这大大改善了救援机器人的自主导航性能。

2 传统 A* 算法

A* 算法在保证搜索最短路径的情况下,通过综合考虑节点到起始点的实际代价和对目标点的估计代价,引入了启发函数的概念,提高路径搜索效率。其实实现过程可分为场景建模与路径规划两个部分。

2.1 场景建模

在 A* 算法中,场景建模模块扮演着至关重要的角色,它能够将实际场景转化为易于计算机读取的栅格地图,从而为路径规划提供了坚实的基础。场景建模方法有栅格法、拓扑法、可视图法和单元分解法等^[6]。与其他建模方法相比,栅格法在操作上更为简单且容易实现,因而被广泛应用于描述机器人环境地图。

在建立地图时,根据实际场景栅格法将环境划分为规则的网格,并在每个网格单元中均存储相应的信息,每一个栅格代表一个节点,这种简单的结构使得栅格法能够更加方便地对相应地图上的区域进行处理与计算。如图 1 所示,白色栅格表示可通行,黑色栅格对应障碍物。映射标记完成后,就可以开始进行路径规划了。

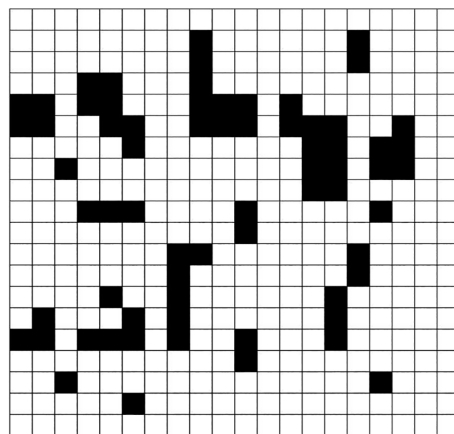


图 1 栅格地图

Fig. 1 Raster map

当移动机器人在栅格地图上进行搜索时,通常会采用 3×3 邻域搜索方式,即以当前节点为中心,从周围的 8 个节点中挑选出可供选择的备选节点,然后从中筛选出下一步将进行扩展的节点。如图 2 所示,为 3×3 邻域的八个运动方向。

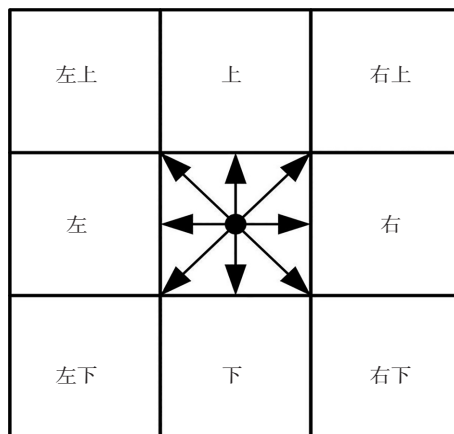


图 2 3×3 邻域

Fig. 2 3×3 neighborhood

2.2 路径规划

传统的 A* 算法利用评价函数来评估从起点到目标点的最短距离,并依据这个评估值来选择接下来要搜索的节点,通过综合考虑实际距离和预估距离,最大限度地提高最短路径的寻找速度。利用评价函数对计算空间中每个节点的总代价进行计算。评价函数如式(1)所示:

$$f(n)=g(n)+h(n)$$
 (1)

式(1)中, n 为当前节点; $g(n)$ 为起点到节点 n 的实际代价; $h(n)$ 为节点 n 到目标点的预估代价; $f(n)$ 为节点 n 的总代价。

传统的 A* 算法用欧氏距离来计算 $h(n)$, 如式(2)、式(3)所示:

$$g(n)=g(f)+\sqrt{(x_n-x_f)^2+(y_n-y_f)^2}$$
 (2)

$$h(n)=\sqrt{(x_g-x_n)^2+(y_g-y_n)^2}$$
 (3)

式(2)、式(3)中, (x_n, y_n) 为当前节点 n 的位置坐标; (x_f, y_f) 为 n 的父节点位置坐标; (x_g, y_g) 为目标点的位置坐标。

通过欧氏距离计算节点到目标点的代价,能够满足最短路径的要求,从而使得传统的 A* 算法在减少搜索节点的同时,也能够像 Dijkstra 算法一样获得最短路径。传统 A* 算法的路径规划具体步骤如算法 1 所示。

算法 1:传统 A* 算法

(1) 创建一个 OpenList 和一个 CloseList,将起点存入 OpenList。

(2) 在当前节点的 3×3 搜索领域中排除 CloseList 中的节点和障碍物节点,筛选出符合条件的其他节点,将它们存入 OpenList。

(3) 对 OpenList 中的节点代价进行比较,选择 f 值最小的作为下一步的扩展节点。

(4) 将扩展过的节点移入 CloseList,如果当前节点不是目标点,则回到第(2)步,继续路径规划;如果是目标点,则完成路径规划。

3 改进 A* 算法设计

3.1 自适应权重

A* 算法中常用曼哈顿距离、对角距离和欧氏距离作为启发函数进行计算。在栅格地图中,移动机器人通常会沿着水平或竖直方向移动,每次移动的距离都是栅格的宽度或高度,而曼哈顿距离则是最直接的距离测量方式。因此,本文采用曼哈顿距离来计算 A* 算法的启发函数,以提高寻路效率。

预估评价启发函数可表示为

$$h(n)=|x_n-x_g|+|y_n-y_g|$$
 (4)

式(1)中,A* 算法的搜索速度和搜索结果很大程度上取决于 $h(n)$ 的选择。当 $h(n)$ 的占比较小时,算法更关注实际代价 $g(n)$,可以保证路径最短,但会扩展冗余节点,导致运行时间增加;而当 $h(n)$ 的占比较大时,算法更关注于搜索靠近目标节点的路径,可以快速找到路径,但无法保证所找到的路径最短。因此,在权衡算法效率与路径优劣之间,动态调整 $h(n)$ 的占比是关键。为了更好地适应移动机器人的路径规划需求,结合移动机器人的位置信息,重新定义总代价函数。具体地,可以将总代价函数表示为

$$f(n)=(1-w)\times g(n)+w\times h(n)$$
 (5)

$$w=1/\left(1+\frac{l}{L}\right)$$
 (6)

式(5)与式(6)中, l 和 L 分别为当前节点到起始节点和起始节点到目标节点的曼哈顿距离。

当距离目标节点较远时, $h(n)$ 权重较大,可以迅速地朝着目标节点前进;靠近目标节点时,两个权重趋近于相同,有助于确保能够抵达目标节点。通过自适应权重的设置,有利于机器人的路径规划,确保在接近目标时能够更加准确地进行导航。这种权重的灵活调整使得路径规划在不同距离情况下都能取得良好的效果。

在不同规格的单位栅格地图中进行路径规划,并对改进前后的算法的搜索性能指标进行比较,最终结果如表 1 所示。可以看出,使用自适应权重的 A* 算法在搜索时间和遍历节点数上都具有较高效率。

表 1 搜索性能指标对比

Table 1 Comparison of search performance metrics

场景大小	算法	搜索时间/ms	路径长度/ 栅格数	遍历节点 数/个
20 * 20	A* 算法	2. 01	29. 8	163
	自适应权重	0. 99	30. 2	78
30 * 30	A* 算法	12. 77	43. 6	241
	自适应权重	6. 99	43. 2	121
50 * 50	A* 算法	17. 27	74. 0	634
	自适应权重	11. 81	74. 0	222

3.2 引入 JPS 跳点策略

传统 A* 算法在搜索过程中,对于每个节点都会直接获取其可达邻居节点作为后继节点,然后对这些后继节点进行评估和扩展,以确保搜索结果的准确性和可靠性。但是这种方式会带来不必要的冗余搜索,尤其是在地图规模增大的情况下,计算复杂度会显著增加。JPS 跳点搜索策略引入跳点这一概念,通过对自然

邻居进行评估,并在路径规划过程中搜索强迫邻居来寻找跳点,从而对无用的节点进行剪枝,优化了邻居节点的搜索方式,从而在大规模地图中减少不必要的搜索,提高了整体路径搜索的效率。相较于直接获取当前节点的可达邻居节点作为后继节点的 A* 算法,该策略的优势更加显著。

3.2.1 自然邻居与强迫邻居

自然邻居的确定与父节点到当前节点的扩展方向有关。具体地,若扩展方向为水平或垂直,则同一方向的下一个节点就是自然邻居;若扩展方向为对角线,那么沿着对角线的下一个节点、下一个垂直的节点以及下一个水平的节点就是自然邻居。如图 3 所示,假设 x 是当前节点, $P(x)$ 是 x 的父节点,图(a)中节点 a 为 x 自然邻居,图(b)中节点 a 、节点 b 和节点 c 为 x 自然邻居。

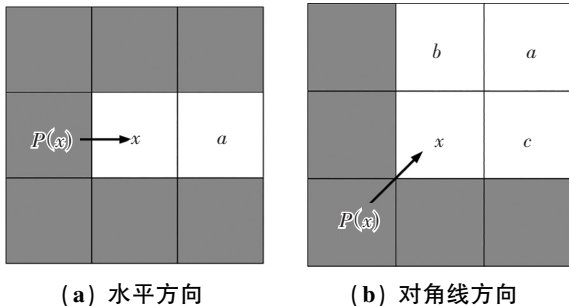


图 3 自然邻居

Fig. 3 Natural neighbors

在节点邻域有障碍物的情况下,当父节点通过节点 x 达节点 n 所需的代价值小于经过其他节点的代价值时,节点 n 为节点 x 的强迫邻居。如图 4 所示,在搜索邻域存在障碍物的时候,强迫邻居节点将会被优先考虑并加入到路径规划中。

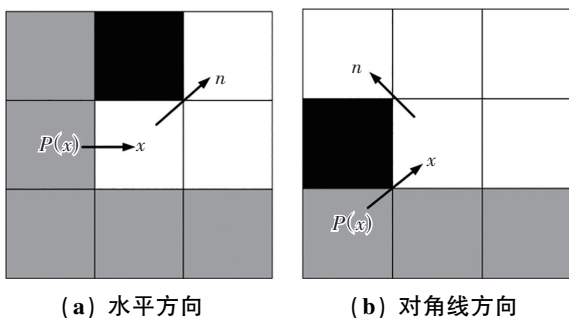


图 4 强迫邻居

Fig. 4 Forced neighbors

3.2.2 跳点筛选规则

在路径规划中,跳点是指通过其他路径无法以最小代价到达的邻居节点。在 JPS 跳点搜索策略中,跳点的存在对于剪枝搜索树、缩小搜索范围及减少冗余计算至关重要。

当 JPS 算法筛选出强迫邻居后,仍需要进一步筛选跳点。因为强迫邻居是在当前节点的某个搜索方向上由于障碍物或边界而被搜索到的节点,而跳点是无法通过其他路径到达的节点。因此,有强迫邻居的地方一定有跳点。

若节点满足以下 3 个条件之一,则可以被判定为跳点:

- (1) 节点为起始点或目标点;
- (2) 节点至少有一个强迫邻居;
- (3) 若父节点到节点为对角线移动,且节点通过水平或垂直的移动可以到达某个跳点时,则节点是跳点。具体示例如图 5 所示, $P(x)$ 为父节点, y 、 z 均为跳点, x 能到达跳点 y ,则 x 为跳点。

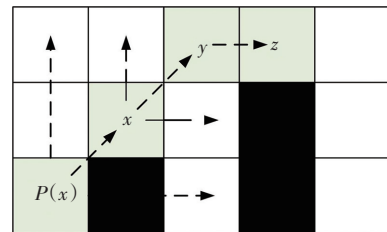


图 5 跳点判断

Fig. 5 Identification of jump points

添加跳点搜索策略的改进 A* 算法具体算法步骤如算法 2 所示。

算法 2:改进 A* 算法

- (1) 创建一个 OpenList 和一个 CloseList,将起点存入 OpenList。
- (2) 在当前节点的 3×3 搜索邻域内,搜索除了 CloseList 中的节点和障碍物以外的跳点,搜索到障碍物时停止,则此方向不被考虑;搜索到跳点并存入 OpenList。
- (3) 当所有方向寻找完毕后,对 OpenList 中的节点进行比较,选出总代价函数值 f 最小的跳点 n 作为当前节点继续寻找路径。
- (4) 将扩展过的节点 n 移入 CloseList,如果当前节点 n 不是目标点,则回到第(2)步,继续路径规划;如果节点 n 是目标点,则完成路径规划。

如图 6 所示,展示了 JPS 跳点搜索策略在路径规划的寻路过程。在图 6 中,蓝色栅格代表开始点,红色栅格代表目标点,而绿色栅格表示经过筛选后保留的跳点。从开始点 S 出发,由于起点的当前方向为空,因此需要在八个方向上扩展来搜索跳点。经过扩展后,发现只有上、右、右上 3 个方向是可行的,于是分别沿着

这3个方向搜索。向上和向右搜索时,遇到边界无法继续扩展,因此只有向右上方向搜索到跳点 T_1 。接着,将跳点 T_1 作扩展节点继续搜索,在 T_1 处,当前方向为水平向右,但此时只能向右搜索,直到向右搜索到跳点 T_2 。在跳点 T_2 处,当前方向为水平方向,沿右、右上2个方向继续搜索,最终到达跳点 T_3 。然后,在跳点 T_3 处,沿着当前方向搜索到达跳点 T_4 ,最后,从跳点 T_4 继续搜索到达目标点 E ,完成了对目标点的搜索。在搜索完成后,通过沿着父节点回溯的方式回到开始点,最终得到了从开始点到目标点的最优路径。

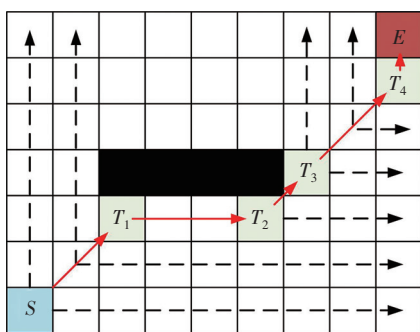


图6 JPS的寻路过程

Fig. 6 Pathfinding process for JPS

3.3 路径平滑

为了确保移动机器人的稳定性,避免出现尖锐拐点对机器人的影响,采用了三次均匀B样条曲线来实现路径的平滑处理。K阶B样条函数表达式为

$$P(u) = \sum_{i=0}^n d_i N_{i,k}(u) \quad (7)$$

式(7)中, i 为节点序号,范围均为从0到 n ; d_i 为控制点坐标; $N_{i,k}(u)$ 是 k 次规范B样条基函数,基函数遵循德布尔考克斯递推式:

$$N_{i,k}(u) = \begin{cases} 1, u_i \leq u < u_{i+1} \\ 0, \text{other}, k=1 \\ \frac{u-u_i}{u_{i+k-1}-u_i} N_{i,k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} N_{i+1,k-1}(u), k \geq 2 \end{cases} \quad (8)$$

式(8)中, k 值的大小决定曲线的平滑度,较大的 k 值会使曲线更加平滑,但同时也增加了计算的复杂性。为了在曲线的平滑度和计算的复杂度之间取得平衡,一般选择 $k=3$,这样能够得到一个平滑度较好的曲线同时保持相对较低的计算复杂性。三次均匀B样条曲线基函数表达式如下:

$$p(u) = p_0 N_{0,3}(u) + p_1 N_{1,3}(u) + p_2 N_{2,3}(u) + p_3 N_{3,3}(u) \quad (9)$$

根据改进算法规划得到的初始路径,将初始返回路径点集作为控制点集 P ,利用式(9)计算得到一系列三次均匀B样条曲线点,然后对这些曲线点进行分段拟合成曲线段,最终将分段曲线组合起来,连接成一条完整的B样条曲线,从而得到平滑后的最终路径。这样的处理过程使得路径在整体上呈现出来连续平滑的特性,为救援机器人的行动提供更加优化和可靠的规划方案,避免机器人运动的不稳定性,提高救援机器人的工作效率。

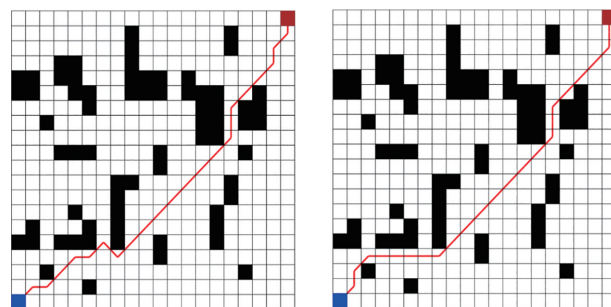
4 仿真实验与结果分析

为了验证将改进自适应权重的A*算法与JPS搜索策略相结合的有效性,在 20×20 、 30×30 与 50×50 大小的地图环境下进行了算法验证。

此外,在Unity3D平台建立 $300 \text{ m} \times 300 \text{ m}$ 的化工园区仿真场景,使用栅格法对地图进行分割,通过仿真实验证明算法的合理性与有效性。

4.1 算法验证

图7—图9展示了不同大小地图的路径规划结果,其中(a)和(b)分别为算法改进前后的规划结果。在栅格地图上的蓝色、红色和黑色方格分别表示起始点、目标点和障碍物。

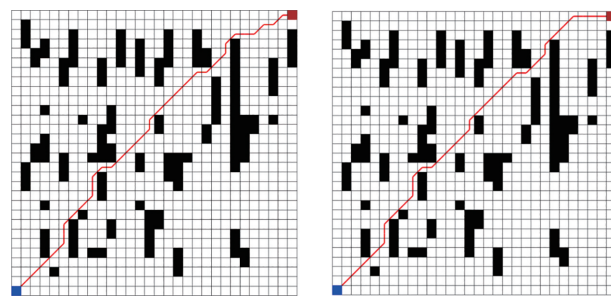


(a) 改进前

(b) 改进后

图7 20×20 仿真结果

Fig. 7 Simulation results of 20×20



(a) 改进前

(b) 改进后

图8 30×30 仿真结果

Fig. 8 Simulation results of 30×30

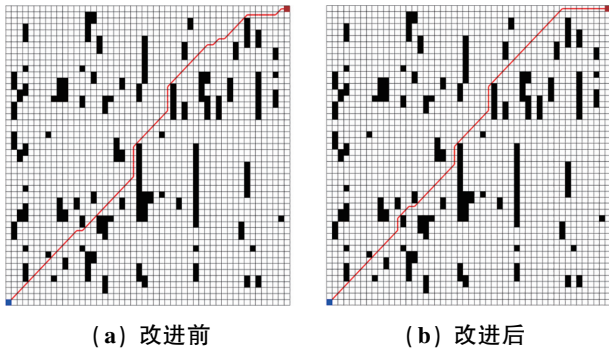


图 9 50×50 仿真结果

Fig. 9 Simulation results of 50×50

通过路径搜索结果可以观察到,虽然传统 A* 算法的路径规划可以找到目标点,但其路径拐点偏多,导致路径不够优化。然而,经过改进的 A* 算法的应用,可以看到规划的路径得到了显著的优化,拐点明显减少,呈现出更平滑的路径。

如表 2 所示,改进的 A* 算法在不同大小的栅格地图都展现出更高效、更智能的路径规划能力,并在大场景地图上具有更大的优势,搜索效率得到了明显提高。

表 2 实验路径参数

Table 2 Experimental path parameters

场景大小	算法	平均搜索 时间/ms	平均路径长 度/栅格数	平均扩展 节点数/个
20 * 20	A* 算法	2.01	29.8	163
	改进 A* 算法	0.95	29.6	37
30 * 30	A* 算法	12.77	43.6	241
	改进 A* 算法	3.61	43.2	53
50 * 50	A* 算法	17.27	74.0	634
	改进 A* 算法	4.93	71.2	88

具体地,改进后平均搜索时间缩短了 65.31%,平均扩展节点减少了 80.47%,能够有效减少冗余节点的搜索,降低内存消耗,提高了搜索效率,从而使算法在复杂环境下表现出更加优异的路径规划性能。

从图 7—图 9 可以看出,规划出的路径均有明显拐点,表现为尖角和折线。使用三次均匀 B 样条曲线消除路径上的拐点,结果如图 10 所示。该方法能够明显地减小因碰撞产生的振动,并且降低了由于障碍物导致机器人运行速度过快造成的震荡现象。通过优化处理,不仅可以提升救援机器人在导航过程中的稳定性和可行性,同时也能有效减少机器人运动时的振动,从

而提高其运动效率。

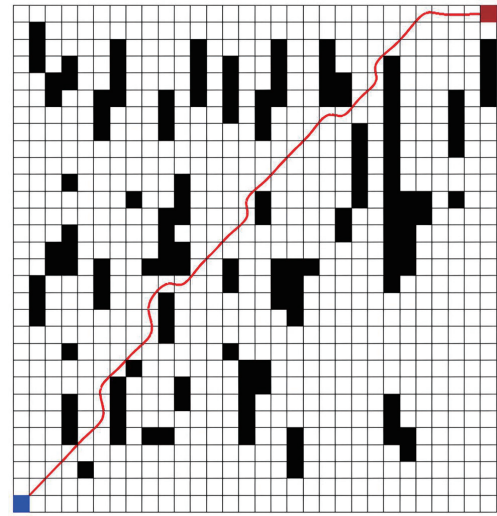


图 10 三次均匀 B 样条曲线路径平滑

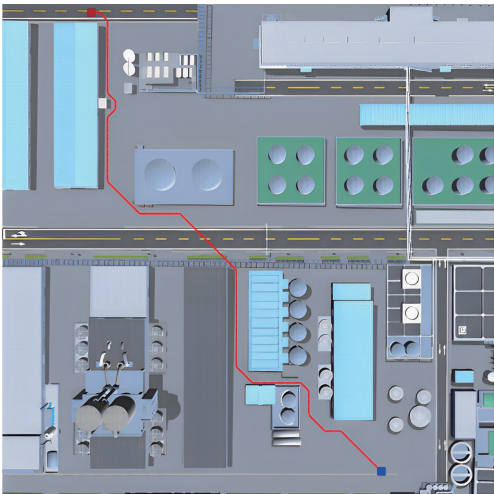
Fig. 10 Cubic uniform B-spline path smoothing

4.2 仿真实验

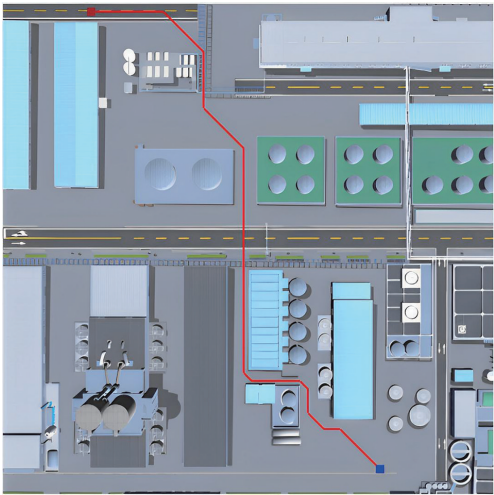
基于化工园区场景,设计仿真实验对改进 A* 算法进行验证。首先,在进行路径规划之前,需要对化工园区进行场景建模,本文使用 3ds MAX 软件搭建化工厂场景模型,将化工园区中的建筑、设施、道路等元素都建模出来。通过精细的场景建模,能够真实地还原化工园区的布局 and 结构。然后,将模型导入到 Unity3D 仿真平台中,可以在仿真环境中很好地模拟出化工园区的真实情况,仿真环境中可以很好地反映实际场景,为路径规划实验提供了一个可靠的基础,以便进行路径规划。

在 300 m×300 m 大小的化工园区地图上,将化工园区场景信息以坐标的形式存入数组,在坐标系中,将左下角设置为原点,并选择起始点坐标为 (230 m, 10 m),目标点坐标为 (50 m, 295 m)。移动机器人在 Unity3D 场景中进行多次对比路径规划仿真实验,规划出的路径以红色线条的形式呈现在地图上,同时起点和终点分别以蓝色方块和红色方块的形式标注出来。

如图 11 所示,是在化工厂场景中机器人使用两种算法完成全局路径规划的效果图。可以看到,改进的 A* 算法规划出的路径更为优化和高效,减少了冗余节点的搜索,路径更加直接,减少了机器人不必要的时间消耗。相较之下,传统 A* 算法规划的路径较为绕远,经过一些不必要的区域,并且存在尖锐拐点,这将增加移动机器人的时间成本。



(a) 改进前



(b) 改进后

图 11 化工园区路径规划

Fig. 11 Path planning for chemical parks

4.3 结果分析

根据两种算法的仿真实验,得到数据如表 3 所示。由数据可知,经过改进的 A* 算法较传统的 A* 算法,平均扩展节点缩短了 79.72%、平均搜索时间减少了 72.12%,同时在搜索效率方面也得到了明显提升。

表 3 改进前后实验结果对照

Table 3 Control of experimental results before and after improvement

指 标	A* 算法	改进A* 算法
平均搜索时间/ms	31.96	8.91
平均路径长度/m	410.37	389.93
平均扩展节点数/个	858	174

规划路径经过平滑优化处理后,其效果如图 12 所示。通过与图 11(b)对比可以看出,平滑处理可以去除

路径中的尖角和折线,使得路径更加自然流畅、平滑,更加符合实际场景的要求,平滑处理还可以降低机器人行驶时的震动和噪声,提高机器人的行驶舒适性。

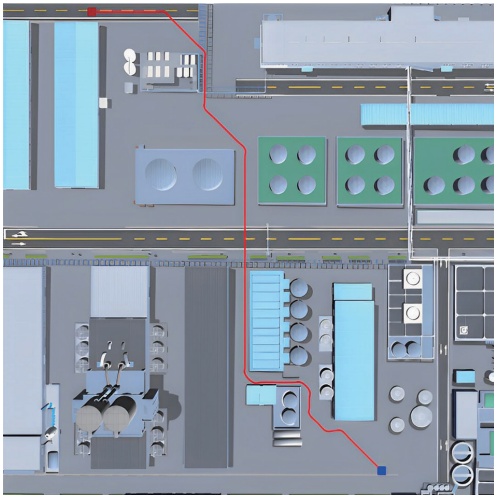


图 12 路径平滑效果

Fig. 12 Path smoothing effect

5 结论与展望

本研究以化工园区救援机器人路径规划为研究对象,提出一种改进的 A* 算法,并通过实验仿真证明该方法的有效性。相较于传统的 A* 算法,该改进算法避免了遍历冗余节点,从而显著降低了内存消耗和计算时间。实验结果表明,该改进 A* 算法在符合实际化工园区情况的条件下,为工业自动化提供了更高效、更智能的路径规划解决方案,具备广泛的实用价值。

本研究在一些方面还有待于深入扩展。目前的研究虽侧重于静态环境下的路径规划,但实际救援任务往往需要多台机器人共同协作才能完成更为复杂的工作。因此,未来的研究可以专注于探索在复杂环境下如何实现多机器人的协同规划。此外,化工园区通常存在着动态的障碍物和环境变化,可以考虑将传感器数据融合到路径规划中,使机器人能够根据环境的变化,自主进行调整,并做出更智能的决策。在后续研究中,将进一步深入探讨这些问题,为实际应用场景提供更加完善和智能的路径规划技术。

参考文献(References):

[1] 万逸飞, 彭力. 改进 A* 蚁群算法求解机器人路径规划问题[J]. 传感器与微系统, 2019, 38(12): 153-156, 160.
WAN Yi-fei, PENG Li. Improved A* ant colony algorithm for robot path planning problem solving [J]. Transducer and Microsystem Technologies, 2019, 38(12): 153-156, 160.

- [2] 李珍惜. 基于激光导航 AGV 路径规划与避障算法的研究[D]. 西安: 西安科技大学, 2019.
- LI Zhen-xi. Research on path planning and obstacle avoidance algorithm based on laser navigation AGV[D]. Xi'an: Xi'an University of Science and Technology, 2019.
- [3] 槐创锋, 郭龙, 贾雪艳, 等. 改进 A*算法与动态窗口法的机器人动态路径规划[J]. 计算机工程与应用, 2021, 57(8): 244-248.
- HUAI Chuang-feng, GUO Long, JIA Xue-yan, et al. Improved A*algorithm and dynamic window method for robot dynamic path planning[J]. Computer Engineering and Applications, 2021, 57(8): 244-248.
- [4] 吴鹏, 桑成军, 陆忠华, 等. 基于改进 A*算法的移动机器人路径规划研究[J]. 计算机工程与应用, 2019, 55(21): 227-233.
- WU Peng, SANG Cheng-jun, LU Zhong-hua, et al. Research on mobile robot path planning based on improved A*algorithm [J]. Computer Engineering and Applications, 2019, 55(21): 227-233.
- [5] 孔继利, 张鹏坤, 刘晓平. 双向搜索机制的改进 A*算法研究[J]. 计算机工程与应用, 2021, 57(8): 231-237.
- KONG Ji-li, ZHANG Peng-kun, LIU Xiao-ping. Research on improved A*algorithm of bidirectional search mechanism [J]. Computer Engineering and Applications, 2021, 57(8): 231-237.
- [6] 罗敏. 复杂环境移动机器人路径规划与轨迹跟踪研究[D]. 成都: 电子科技大学, 2021.
- LUO Min. Research on path planning and trajectory tracking of mobile robot in complex environment[D]. Chengdu: University of Electronic Science and Technology of China, 2021.
- [7] 金震, 黄卫华, 李传奇, 等. 基于改进 JPS 和 A*算法的组合路径规划[J]. 高技术通讯, 2022, 32(4): 412-420.
- JIN Zhen, HUANG Wei-hua, LI Chuan-qi, et al. Combination path planning based on improved jump point search and A* algorithm [J]. Chinese High Technology Letters, 2022, 32(4): 412-420.
- [8] 苗红霞, 郭章旺, 齐本胜, 等. 基于并行-交替式双向 JPS 算法的机器人路径规划[J]. 计算机测量与控制, 2022, 30(7): 233-239.
- MIAO Hong-xia, GUO Zhang-wang, QI Ben-sheng, et al. Robot path planning based on parallel alternate bidirectional JPS algorithm[J]. Computer Measurement & Control, 2022, 30(7): 233-239.
- [9] 王远国, 俞海兵, 李云臣, 等. 基于改进人工蜂群算法的核电巡检机器人路径优化策略设计[J]. 电子设计工程, 2023, 31(6): 70-74.
- WANG Yuan-guo, YU Hai-bing, LI Yun-chen, et al. Path optimization strategy design of nuclear power inspection robot based on improved artificial bee colony algorithm[J]. Electronic Design Engineering, 2023, 31(6): 70-74.
- [10] ELBANHAWIM, SIMIC M, JAZAR R. Randomized bidirectional B-spline parameterization motion planning[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(2): 406-419.
- [11] WANG Y, ZHANG S, WU Y, et al. Research on path planning of mobile robot based on improved jump point search algorithm[C]//Proceedings of the IEEE International Conference on Mechatronics and Automation. Piscataway: IEEE Press, 2021: 101-106.
- [12] LAN X, LV X, LIU W, et al. Research on robot global path planning based on improved A-star ant colony algorithm[C]//Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference. Piscataway: IEEE Press, 2021: 613-617.
- [13] 姜媛媛, 张阳阳. 改进 8 邻域节点搜索策略 A*算法的路径规划[J]. 电子测量与仪器学报, 2022, 36(5): 234-241.
- JIANG Yuan-yuan, ZHANG Yang-yang. Improved path planning of A*algorithm of domain node search strategy 8[J]. Journal of Electronic Measurement and Instrumentation, 2022, 36(5): 234-241.
- [14] 于洋, 周佳伟, 冯迎宾, 等. 基于三次 B 样条曲线的无人车轨迹优化方法研究[J]. 沈阳理工大学学报, 2019, 38(5): 71-75.
- YU Yang, ZHOU Jia-wei, FENG Ying-bin, et al. Research on trajectory optimization of unmanned vehicle based on cubic B-spline interpolation[J]. Journal of Shenyang Ligong University, 2019, 38(5): 71-75.
- [15] 童雨聪. 基于激光雷达的室内移动机器人导航研究[D]. 武汉: 武汉科技大学, 2021.
- TONG Yu-cong. Research on indoor mobile robot navigation based on lidar[D]. Wuhan: Wuhan University of Science and Technology, 2021.
- [16] 肖献强, 王鼎用, 王家恩, 等. AGV 复合自主路径规划方法研究[J]. 机械设计与制造, 2022(3): 262-265.
- XIAO Xian-qiang, WANG Ding-yong, WANG Jia-en, et al. A research on AGV composite autonomous path planning method [J]. Machinery Design & Manufacture, 2022(3): 262-265.

责任编辑:陈 芳